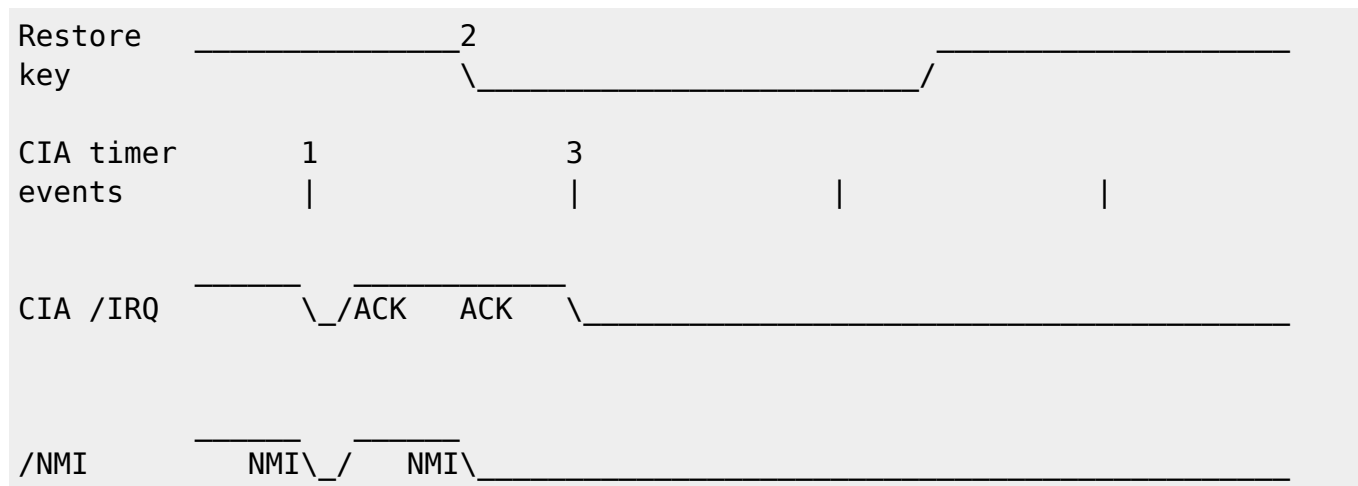


Surviving Restore key presses while using CIA 2 timer NMIs

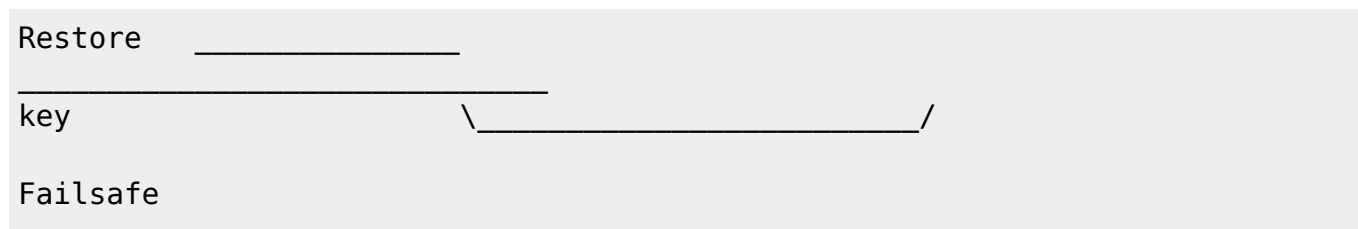
The /NMI pulse from the restore key seems to be longer than 1 frame. When this happens, the CIA 2 interrupt source doesn't cause the NMI to trigger, so the timer never gets ACKed, and NMIs appear to stop as /NMI is locked low, even though the timer is still running.

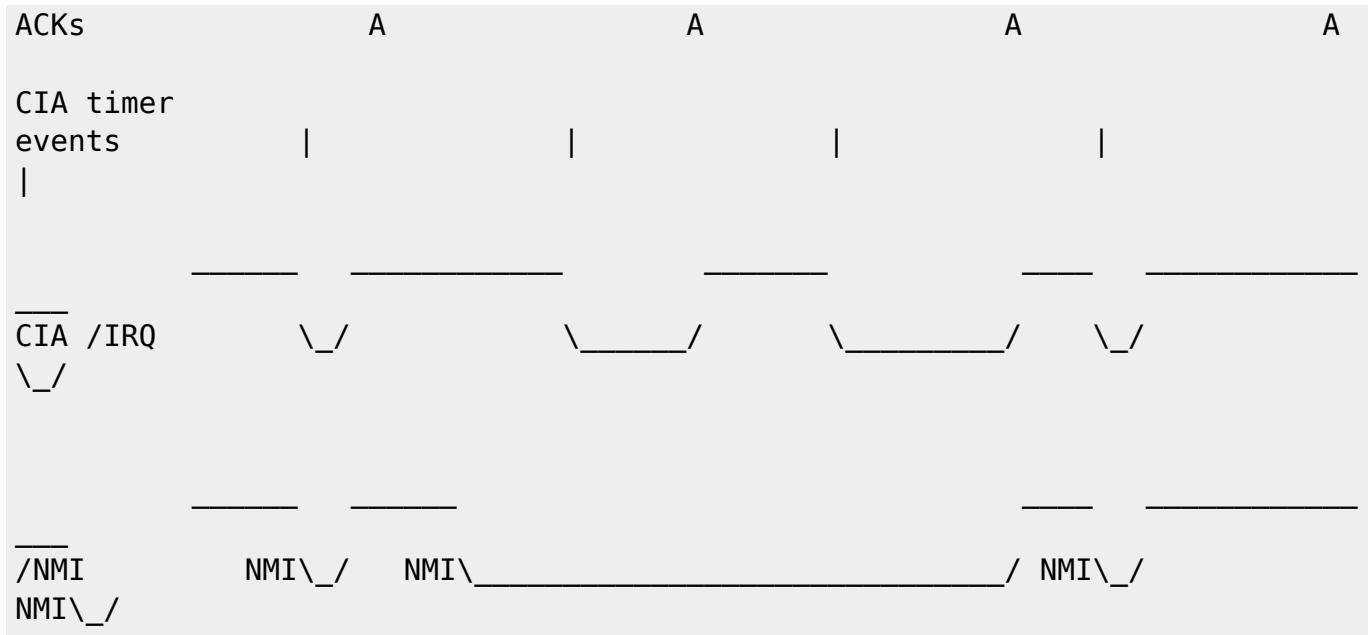


At location 1, a normal timer interrupt occurs, drawing /NMI low. The processor runs the NMI handler, which ACKs the timer, releasing the /NMI line to its default high state. When the Restore key is pressed at location 2, the NMI handler is called and can detect that the CIA did not cause the NMI. However, NMI is not released by the time the next timer event occurs. This event will ground the /NMI line, but it's already still grounded from the Restore key, so the processor doesn't see the falling-edge event. The CIA timer keeps running, but since one of the timer events was never ACKed, the /NMI stays low indefinitely.

There are a couple of ways to resolve this. The simplest one is simply to periodically ACK the CIA timer from the main non-interrupt code, which will eventually release the CIA's /NMI assertion if a timer ACK is missed due to the Restore key. A few timer events will have been lost, but the CPU will start to receive NMI events again after the Restore key pulse has ended. When the timer is periodically ACKed by the main code, it can check to see if the timer interrupt was actually asserting, and trigger code to handle the lost-event case when the NMI comes back online.

The redundant ACKs will not interfere with NMI triggering. However, there is the slight possibility that they will read the CIA interrupt register (thus clearing it) as the NMI occurs, in which case the NMI handler will think that the CIA didn't trigger it. If there is only one NMI input source (disregarding the Restore key) and the failsafe ACKs are happening, you shouldn't have to read the bits of the interrupt register. If there are multiple input sources and they must be distinguished, keeping a more complex watchdog routine in the main code would be required that ACKs the timer only when it notices that NMIs are not occurring.





-
White Flame (aka David Holz)
<http://www.white-flame.com/>

From:
<https://codebase64.org/> - **Codebase 64** wiki

Permanent link:
https://codebase64.org/doku.php?id=base:restore_and_cia_2&rev=1429238017

Last update: **2015-04-17 04:33**

